

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 0 918 400 A2

(12)

## EUROPEAN PATENT APPLICATION

(43) Date of publication:

26.05.1999 Bulletin 1999/21

(51) Int. Cl.<sup>6</sup>: H04B 1/66

(21) Application number: 98120786.3

(22) Date of filing: 03.11.1998

(84) Designated Contracting States:

AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE

Designated Extension States:

AL LT LV MK RO SI

(30) Priority: 10.11.1997 JP 307394/97

(71) Applicant:

Matsushita Electric Industrial Co., Ltd.  
Kadoma-shi, Osaka-fu, 571-8501 (JP)

(72) Inventor: Bett, Russell

Hampshire, S018 5TB (GB)

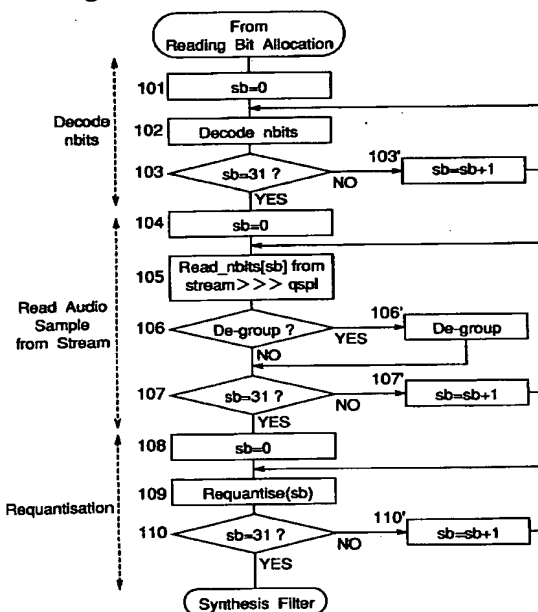
(74) Representative:

Eisenführ, Spelser & Partner  
Martinistrasse 24  
28195 Bremen (DE)

(54) Method for decoding coefficients of quantization per subband using a compressed table

(57) Method disclosed is used for compressing the four quantization per subband tables used in MPEG1 layer 2 audio to minimize required memory size. The method also describes how to decode the compressed tables and use the data to maximize efficiency when reading the audio samples from the input stream, de-grouping and requantization of the audio samples. The efficiency of this method is suitable for use with the implantation of RISC/micro-processor based MPEG1 layer 2 audio decoder.

Fig. 1



EP 0 918 400 A2

## Description

## BACKGROUND OF THE INVENTION

## 1. Field of the Invention

[0001] The present invention relates to the implementation of linear quantization for use in digital audio encoder and decoder applications.

## 2. Description of the Related Art

[0002] When coding an analogue signal to a digital representation, the values of the analogue signal which can be encoded are restricted to a number of values or quantized steps. For efficient digital encoding, the number of bits representing the audio samples will vary in number. The number of bits per audio sample are stored in a table, being quantization per subband.

[0003] MPEG1 (Layer 2) uses four such quantization tables, only one table is used depending on the sampling frequency and bitrate. Each table has sixteen columns and contains data for each of the thirty two subbands. The memory required to store the uncompressed data is 2048 x 2 bytes.

[0004] There are several methods of compressing the four quantization tables, one method is to compress each tables row of data. Repeated rows are not encoded. Also, if other quantization tables comprise of the same row information as one previously compressed, than this table need not be encoded. In this manner, the four quantization tables used in MPEG1 (Layer 2) can be compressed into two tables. However, to recreate the original tables, an additional four tables are required. These comprise of thirty two values used to point to the start location of the row data stored in one of the two previously compressed tables. This compression method requires memory space for 776 x 2 bytes.

[0005] There are several problems to be solved. The need for better quality digital audio has prompted the increase of quantization levels and thus quantization tables. When designing an audio decoder in LSI using a RISC based processor, the cost of the chip increases with increase in size. A major factor contributing to the chip size is the amount of memory used. When implementing a software method of de-compression, the processing load increases.

## SUMMARY OF THE INVENTION

[0006] The object of this invention is to provide an efficient method of compression requiring minimal decompression time.

[0007] Each of the said quantization per subband tables are compressed into tables containing a number of parts. Each of the said parts being the compressed format of one column. The said parts consist of three pairs of data. The first data value of the pair define the subband up to which the said nbits value remains the same, here after known as subband marker. For the first pair, the said nbits remains constant for all subbands from zero to the said subband marker. Second and third subband markers refer to the range from that of the previous subband marker to that of the current subband marker. If the current subband number is less than or equal to the subband marker, than the said nbits is equal the second value of the pair. Each pair of data are examined until the said nbits has been found or, all three pairs have been searched. If the said nbits is not found, nbits is by default equal zero.

[0008] The subband marker of the third pair need not be equal the last subband. For all subbands greater than the value of the subband marker in the last pair the said nbits is equal zero.

[0009] In the case of MPEG1 (layer 2) audio decoder, the invention eliminates memory wastage by encoding each of the original quantization per subband tables each into one part containing three pairs of data. For subbands up to and including the first value of the pair, the number of bits representing the audio sample (nbits) is equal the second value of the pair. The second pair is decoded such that, for subbands greater than that of the first subband marker up to and including the second subband marker, the nbits is equal the second value of the second part. Similarly for the third part, all subbands greater than the second subband marker up to and including the third subband marker, the nbits is equal the second value of the third pair. Each pair is tested until the quantization has been determined. Should the said nbits remain unresolved after examining the third pair, said nbits is equal zero.

[0010] The quantization data is decoded for each of the subbands using sb as the counter. The appropriate part being indexed by multiplying the bit allocation data by six. The quantization data for all subbands are stored in said \_nbits.

[0011] Encoding operation is as follow.

[0012] After identifying the nominal number of bits required to store an audio sample. If the samples are similar in amplitude, than grouping may be implemented. If so, this is signified by setting bit five of nbits high. After identifying nbits, the quantized sample is applied to a linear formula.

[0013] As stated in ISO/IEC 11172-3:1993(E), the subband sample is quantized by applying the value to the following

linear formula:

$$Q = AX + B$$

5 [0014] X, being the subband sample divided by the scale factor.

[0015] The constants A and B are read from a table, quantization coefficients. A and B are indexed by nbits. If nbits equal three, then nbits is equal zero. Should nbits be greater than sixteen, the indexing value is equal the least significant two bits of nbits. Otherwise, the indexing value is equal nbits. After generating the index value, the A and B values are read and the linear formula executed.

10 [0016] Decoding operation is as follows.

[0017] After storing all quantization data, the quantized audio samples can be read from the bit stream. The number of bits to be read for each quantized audio sample is equal nbits, nbits being read from \_nbits indexed by sb from zero to thirty one. The number of bits to be read is equal the value of the least significant four bits of nbits. Should the number of bits read from memory be greater than sixteen, then the quantized audio sample must be regrouped to three quantized audio samples. The quantized audio samples for each subband are stored in memory.

15 [0018] As stated in ISO/IEC 11172-3 1993(E), the quantized audio samples are dequantized by means of a linear formula:

$$S[sb] = C \times (D + qspl[sb])$$

20 [0019] The constants C and D are read from a table, classes of quantization. C and D are indexed by nbits for subband sb. If nbits equal three, then nbits is equal zero. Should nbits be greater than sixteen, the indexing value is equal the least significant two bits of nbits. Otherwise, the indexing value is equal nbits. After generating the index value, the C and D values are read and the linear formula executed.

25 [0020] A first aspect of the invention is a method to compress tables of linear quantization data used to digitally represent analogue signals, each table comprising the structure of:

a number of columns being coded into groups of data;  
the number of said columns is dependent on the accuracy of the coding and frequency bandwidth of the original  
30 signal, hereafter known as allocation;  
length of said columns being equal the number of sub divisions used to split the audio frequency band, hereafter known as subbands; and  
three pairs of data for means of compressing each said column.

35 [0021] A second aspect of the invention is a method of compressing said columns according to the first aspect, comprising of the following steps:

comparing said subband to the first value of the said pair hereafter known as subband marker;  
when the said subband is less than the said subband marker, the number of bits used to quantize the sample is  
40 equal the second value of the said pair, hereafter being known as nbits;  
repeating the comparison of the three said pairs is done until said nbits has been found; and  
following the comparison of all three said pairs, the value of said nbits may not be resolved in such a situation, said nbits is equal zero.

45 [0022] A third aspect of the invention is a method implementing said compressed quantization tables according to first and second aspects, for the encoding process comprising the steps of:

selecting the compressed quantization per subband table;  
selecting a suitable said allocation value for each sample;  
50 multiplying the said allocation data value by six to index the said compressed quantization table, the indexed value being nbits;  
comparing the noise incurred by coding the analogue sample using said nbits to that of the acceptable noise level generated from the psychoacoustic model; and  
repeating the selection process of said allocation to acquire the least number of said nbits needed to represent the  
55 analogue sample at the optimum acceptable noise level.

[0023] A fourth aspect of the invention is a method using said allocation to read said nbits from said compressed quantization tables in accordance to first and second aspects, comprising the steps of:

reading said allocation for each and every subband;  
 storing the said allocation to memory hereafter known as alc;  
 using a counter, known hereafter as sb, to identify subband number  
 selecting the said compressed quantization per subband table;  
 5 reading said allocation, being said alc memory indexed by said sb;  
 calculating pointer to said compressed quantization per subband table using said allocation multiplied by six;  
 comparing data being indexed by said pointer to said compressed quantization per subband table with said sb  
 (being said subband marker);  
 should said sb be less than or equal to said subband marker, said nbits representing the analogue sample, is equal  
 10 value being indexed by said pointer plus one to said compressed quantization per subband table;  
 comparing said sb to all three said parts until said nbits has been found;  
 if, after comparing all three said parts, said nbits has not been found, said nbits is equal zero;  
 said nbits is stored to memory, hereafter being known as \_nbits, being indexed by said sb;  
 said nbits are decoded for all subbands;  
 15 reading quantized sample data from the bitstream, hereafter known as qspl, the number of bits representing said  
 qspl being equal said nbits read from said \_nbits being indexed by said sb; and  
 reading of said qspl is repeated for all said subbands.

[0024] A fifth aspect of the invention is a method to quantize/requantize samples comprising of a linear applying the  
 20 samples to a linear formula, comprising the steps of:

reading said nbits for the current said subband from said \_nbits being indexed by said sb;  
 decoding said nbits, hereafter to be known as linear.ptr;  
 reading linear constants being indexed by said linear.ptr; and  
 25 applying said linear constants and sample to said linear formula.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0025]

30 Figure 1 is a flow chart giving an overview of when the processes to decode the compressed quantized levels per  
 subband tables to nbits, using nbits to read the audio samples from the input data stream and repquantization of the  
 audio samples read.

Figure 2 is a flow chart illustrating the decoding and generation on nbits per subband.

35 Figure 3 shows a flow chart showing the processes involved in the requantization of the audio samples.

Figure 4 shows a flow chart to illustrate the use of nbits to index the re-ordered classes of quantization table.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

40 [0026] An example of the embodiment will be discussed hereafter. First, a general picture of implementation is  
 explained.

[0027] Figure 1 shows the use of compressed quantization tables and data. With reference to Figure 1, this diagram  
 highlights a small section of the decoding process. First, the number of bits (nbits) to be read from the stream per audio  
 sample is decoded. The process is repeated for each subband, thus using a subband loop pointer, sb, being initialized  
 45 to zero (step 101). The decoding process (step 102) implements the compressed quantization per sample tables, nbits  
 is stored in array \_nbits, indexed by sb pointer. Following the decoding of nbits for each subband, the subband pointer,  
 sb is checked if all subbands have been decoded (step 103) and if not the sb pointer is incremented (step 103') and the  
 process repeated. The implementation of this process will be discussed later.

[0028] The next stage is to read the audio samples from the stream (step 105), the number of bits read is equal the  
 50 value in \_nbits, being indexed by the current sb pointer. After reading nbits, if the value is greater than sixteen, the sam-  
 ple read from the stream must be un-grouped (steps 106 and 106'). The audio samples are stored as \_qspl, being  
 indexed by sb pointer. Similarly, this process is repeated for all subbands thus looping until sb pointer is equal thirty one  
 (steps 105-107'). The reading of audio samples and de-grouping can be executed as described in ISO/IEC 11172-  
 3:1993 (E) and will not be described any further.

55 [0029] After reading and storing the quantized audio samples (qspl), the audio samples must be dequantized (step  
 109). Again, this process is repeated for all subbands, thus using a sb pointer (steps 109-110'). The dequantization  
 process will be discussed in greater detail later.

[0030] Next, the decoding of the number of bits per audio sample is described.

# EP 0 918 400 A2

[0031] Figure 2 shows a flow chart for decoding number of bits per subband sample. With reference to Figure 2, in order to determine which of the four compressed tables to decode, first the sampling frequency and bitrate (step 201) must be decoded from the audio stream header. This information is used to determine the value of sb limit and thus which of the four compressed quantization per subband tables to use, Table 1, Table 2, Table 3 and Table 4.

Table 1

Compressed 2Ba Quantization per subband					
sb pointer	quant.	sb pointer	quant.	sb pointer	quant.
31	0	31	0	31	0
26	21	31	0	31	0
2	3	26	23	31	0
2	4	22	3	26	16
2	5	22	26	31	0
2	6	22	4	31	0
2	7	22	5	31	0
2	8	10	6	22	16
2	9	10	7	31	0
2	10	10	8	31	0
2	11	10	9	31	0
2	12	10	10	31	0
2	13	10	11	31	0
2	14	10	12	31	0
2	15	10	13	31	0
10	16	31	0	31	0

Table 2

Compressed 2Bb Quantization per subband					
sb pointer	quant.	sb pointer	quant.	sb pointer	quant.
31	0	31	0	31	0
29	21	31	0	31	0
2	3	29	23	31	0
2	4	22	3	29	16
2	5	22	26	31	0
2	6	22	4	31	0
2	7	22	5	31	0
2	8	10	6	22	16
2	9	10	7	31	0
2	10	10	8	31	0
2	11	10	9	31	0

# EP 0 918 400 A2

Table 2 (continued)

Compressed 2Bb Quantization per subband					
sb pointer	quant.	sb pointer	quant.	sb pointer	quant.
2	12	10	10	31	0
2	13	10	11	31	0
2	14	10	12	31	0
2	15	10	13	31	0
10	16	31	0	31	0

Table 3

Compressed 2Bc Quantization per subband					
sb pointer	quant.	sb pointer	quant.	sb pointer	quant.
31	0	31	0	31	0
7	21	31	0	31	0
7	23	31	0	31	0
7	26	31	0	31	0
7	4	31	0	31	0
7	5	31	0	31	0
7	6	31	0	31	0
7	7	31	0	31	0
1	8	31	0	31	0
1	9	31	0	31	0
1	10	31	0	31	0
1	11	31	0	31	0
1	12	31	0	31	0
1	13	31	0	31	0
1	14	31	0	31	0
1	15	31	0	31	0

Table 4

compressed 2Bd Quantization per subband					
sb pointer	quant.	sb pointer	quant.	sb pointer	quant.
31	0	31	0	31	0
7	21	31	0	31	0
11	23	31	0	31	0
11	26	31	0	31	0

Table 4 (continued)

compressed 2Bd Quantization per subband					
sb pointer	quant.	sb pointer	quant.	sb pointer	quant.
11	4	31	0	31	0
11	5	31	0	31	0
11	6	31	0	31	0
11	7	31	0	31	0
1	8	31	0	31	0
1	9	31	0	31	0
1	10	31	0	31	0
1	11	31	0	31	0
1	12	31	0	31	0
1	13	31	0	31	0
1	14	31	0	31	0
1	15	31	0	31	0

[0032] The selected table hereafter will be referred to as quant.tbl. The number of bits per audio sample is decoded for each of the thirty two subbands using a loop counter called sb being initialized to zero (step 202). The pointer to quant.tbl (step 203) is generated by reading the bit allocation data for the current subband and multiplying it by six since there are six data for each subband. The subband marker being indexed by the previously derived pointer, is compared with the current subband counter, sb (step 204). If sb is less than or equal to the value being indexed, the number of audio bits to be read, nbits, is equal the value indexed by the pointer plus one (step 212). However, if sb is greater than the indexed subband marker, the pointer is incremented by two (step 205) in order to point to the next pair of data. Similarly, the current sb is compared to the subband marker (step 206). Again, if the current sb is less than or equal to the subband marker, nbits is equal quant.table indexed by the pointer plus one. If sb is greater than the subband marker, the pointer is incremented by two (step 207) before comparing the next pair. Similarly, if the current sb is less than or equal to the final subband marker (step 208), nbits is assigned the value of quant.tbl, indexed by pointer plus one. Since this is the final part, if sb is greater than the subband marker, nbits is set to zero (step 209). The value nbits is stored in memory array \_nbits (step 210), this being indexed by sb. The decoding of nbits is executed for all thirty two subband samples, therefore, if sb is not equal thirty one (step 211), sb is incremented by one (step 211') and the decoding process repeats from (step 203) onward.

[0033] Next, dequantization is described.

[0034] Figure 3 is a flow chart for dequantization of quantized audio samples. After the decoding of audio samples (qspl) from the bit stream, the samples must be dequantized. The dequantization process is carried out in three sections, with reference to Figure 3.

[0035] Firstly, (step 302) the most significant bit of qspl is inverted thus producing a two's complement fractional number.

[0036] Secondly, a linear formula is applied to the new value of qspl (step 303), this is the only section which differs from the original specification (ISO/IEC 11172-3:1993(E)) and will be described in greater detail later.

[0037] The final part is to scale the requantized value (step 304). The requantization process is executed on all subband samples, thus implementing a subband pointer (sb) first being initiated to zero (step 301) and compared after the requantization of each sample (step 305).

[0038] A linear formula is described next.

[0039] The linear formula is as specified in ISO/IEC 11172-3:1993(E):

$$S = C \times (D + s^n)$$

[0040] The process of execution differs from that of the original specification by the process of finding the values of C and D, these are read from the modified table 5. Modified, meaning that the order of rows has been changed to make decoding simpler.

Table 5

Layer II Classes of Quantization				
Number of steps	Grouping	Samples per code word	Bits per code word	Encoded quantization
7	no	1	3	3
3	yes	3	5	21
9	yes	3	10	26
5	yes	3	7	23
15	no	1	4	4
31	no	1	5	5
63	no	1	6	6
127	no	1	7	7
255	no	1	8	8
511	no	1	9	9
1023	no	1	10	10
2047	no	1	11	11
4095	no	1	12	12
8191	no	1	13	13
16383	no	1	14	14
32768	no	1	15	15
65535	no	1	16	16

**[0041]** Figure 4 is a flow chart of decoding C and D index pointer. With reference to figure 4, C and D are read from the reordered table by a pointer (C.D.ptr), the pointer is initiated to nbits for the current subband (step 401). Should C.D.ptr be less than or equal to sixteen (step 402), the current C.D.ptr value stands valid. However, if C.D.ptr is three (step 403), C.D.ptr becomes zero (step 404). In the case when, C.D.ptr is greater than sixteen, only the least significant two bits of C.D.ptr are valid (step 405). The C and D values can now be read from table 5 using the C.D.ptr as the index.

**[0042]** Clearly, from the above description of this novel table compression technique, implementing this method reduces the amount of memory required to store and decode the quantization levels per subband. The tables are compressed to 384 x 2Bytes data values being decoded to 32 data values. Not only is this compression method efficient on memory but also the decoding process is also efficient on processing time.

#### Claims

1. A method to compress tables of linear quantization data used to digitally represent analogue signals, each table comprising the structure of:

a number of columns being coded into groups of data;  
the number of said columns is dependent on the accuracy of the coding and frequency bandwidth of the original signal, hereafter known as allocation;  
length of said columns being equal the number of sub divisions used to split the audio frequency band, hereafter known as subbands; and  
three pairs of data for means of compressing each said column.

2. A method of compressing said columns according to claim 1, comprising of the following steps:

comparing said subband to the first value of the said pair hereafter known as subband marker;  
when the said subband is less than the said subband marker, the number of bits used to quantize the sample

is equal the second value of the said pair, hereafter being known as nbits;  
 repeating the comparison of the three said pairs is done until said nbits has been found; and  
 following the comparison of all three said pairs, the value of said nbits may not be resolved in such a situation,  
 said nbits is equal zero.

- 5  
 3. A method implementing said compressed quantization tables according to claims 1 and 2, for the encoding process comprising the steps of:

10 selecting the compressed quantization per subband table;  
 selecting a suitable said allocation value for each sample;  
 multiplying the said allocation data value by six to index the said compressed quantization table, the indexed value being nbits;  
 comparing the noise incurred by coding the analogue sample using said nbits to that of the acceptable noise level generated from the psychoacoustic model; and  
 15 repeating the selection process of said allocation to acquire the least number of said nbits needed to represent the analogue sample at the optimum acceptable noise level.

4. A method using said allocation to read said nbits from said compressed quantization tables in accordance to claims 1 and 2, comprising the steps of:

20 reading said allocation for each and every subband;  
 storing the said allocation to memory hereafter known as alc;  
 using a counter, known hereafter as sb, to identify subband number;  
 selecting the said compressed quantization per subband table;  
 25 reading said allocation, being said alc memory indexed by said sb;  
 calculating pointer to said compressed quantization per subband table using said allocation multiplied by six;  
 comparing data being indexed by said pointer to said compressed quantization per subband table with said sb (being said subband marker);  
 should said sb be less than or equal to said subband marker, said nbits representing the analogue sample, is equal value being indexed by said pointer plus one to said compressed quantization per subband table;  
 30 comparing said sb to all three said parts until said nbits has been found;  
 if, after comparing all three said parts, said nbits has not been found, said nbits is equal zero;  
 said nbits is stored to memory, hereafter being known as \_nbits, being indexed by said sb;  
 said nbits are decoded for all subbands;  
 35 reading quantized sample data from the bitstream, hereafter known as qspl, the number of bits representing said qspl being equal said nbits read from said \_nbits being indexed by said sb; and  
 reading of said qspl is repeated for all said subbands.

5. A method to quantize/requantize samples comprising of a linear applying the samples to a linear formula, comprising of the following steps:

40 reading said nbits for the current said subband from said \_nbits being indexed by said sb;  
 decoding said nbits, hereafter to be known as linear.ptr;  
 reading linear constants being indexed by said linear.ptr; and  
 45 applying said linear constants and sample to said linear formula.

Fig. 1

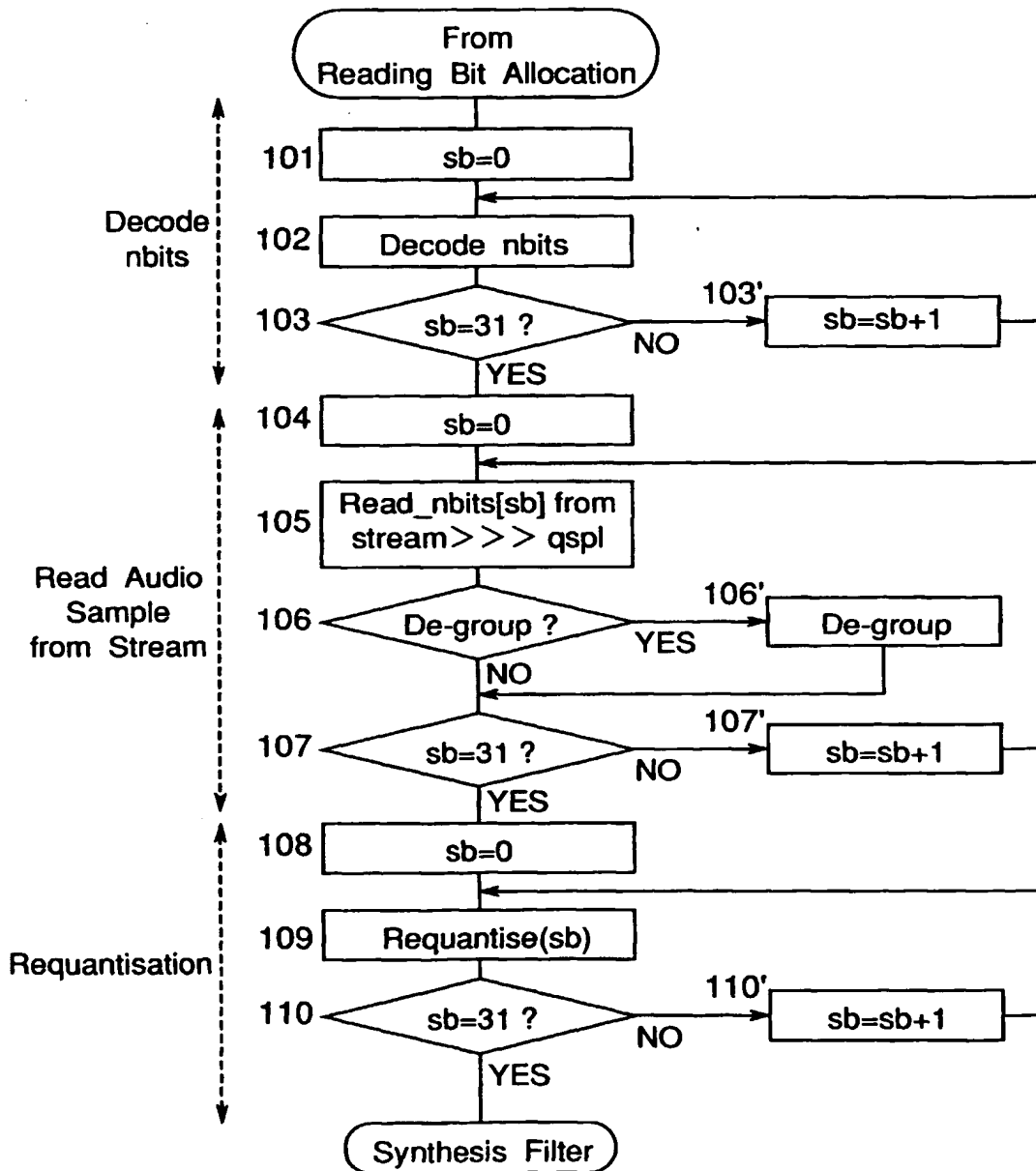
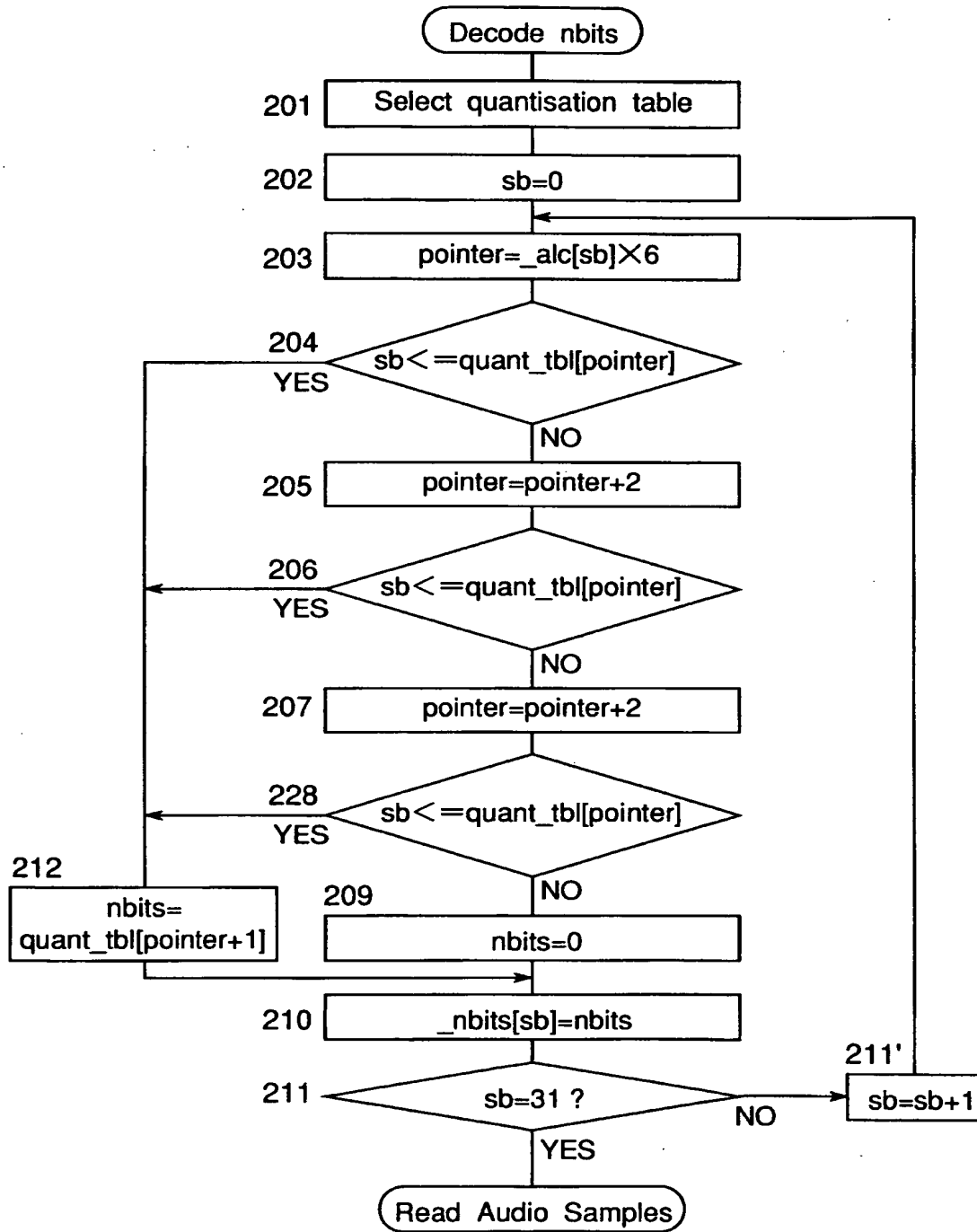
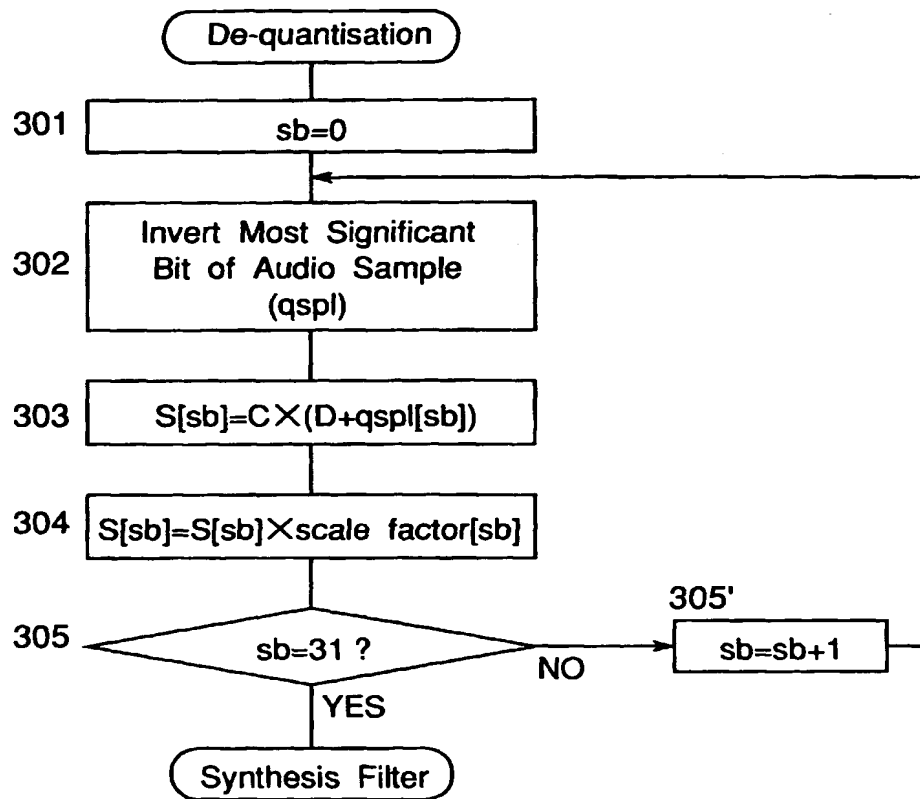


Fig.2



*Fig.3*

*Fig.4*